REMARKS

Claims 1-10, 12, 14-21 and 23 remain in the application. No Claim has been allowed.

Claim 1 and various other claims now stand rejected under 35 U.S.C. § 103(a) as being unpatentable in view of a single prior art reference, Blumenau et al. (U.S. Patent Publication 2004/0080558). [1]

By way of review, the Applicants' invention is directed to approaches for controlling resource distribution in a server-based, partitioned, resource storage system. In such a system, a plurality of storage servers store a set of partitioned resources. That is, portions of a given resource are stored on one server, and other portions of that same resource are stored on other servers elsewhere in the system. See for example, Fig. 4 and paragraph 44 et seq. in Applicants' disclosure. The storage servers each have a respective load monitor processes that communicates with the load monitor processes in the other servers to determine a measure of loading at each respective storage server. Based on the determined measure of loading, portions of the partitioned resources are transferred back and forth, from one server to another. A write detect process detects when a resource is in the process of being moved ("migrated") between a source server (the "first server") and a second server (the "target server").

The claims are specifically directed to what happens when a write failure occurs on the target server (i.e., the "second server" as recited in the claims). This is handled in a simple and efficient way – by restarting the migration process for the resource which had the write request issued to it. In this way, the simultaneous write to the source server (i.e., the claimed first server) is leveraged. The effected resource is simply re-migrated, "ensuring that the write request is propagated to the second server," as claimed.

As has been noted in Applicants' previous correspondence, Blumenau does not disclose an approach for detecting target server write failures in a resource migration process, and in response thereto, restarting that same migration process in a partitioned resource storage system.

We note first that Blumenau discloses a much more involved and complicated data recovery process that compares the source and target data with state information to determine

---

[1] This is despite the fact that a rejection under 35 U.S.C. §103 in view of this same Blumenau prior art in combination with three other references (Mashayekhi, Umberger and Aditya) has now been withdrawn. If Blumenau in combination with the other prior art does not render the claims obvious, certainly Blumenau alone cannot.

which data is "good" and which is "bad" (see paragraph [0048], lines 2-5 and paragraph [0049], lines 2-7). For example, the state information may be a count which indicates the number of data operations performed on a particular storage location [0051]. Blumenau's recovery process then specifically copies the good data from the storage location where the write completed successfully to the other location, based on the state information. See paragraph [0048], lines 5-7 and [85], lines 9-12. The data in the location where the most recent write occurred is thus relied upon as being the "good" data, based on the state information. Alternatively, Blumenau's recovery process invalidates data stored at both locations. See paragraph [0048], lines 20-23 and paragraph [85], lines 9-11. Blumenau's process is thus entirely different from Applicants simplified approach of restarting the migration process for the resource in response to a write failure on the second ("target") server.

The Examiner points to Blumenau at paragraphs [0039 through 0040] as supposedly teaching this feature of Applicants' claimed invention, of writing to both source and target volumes in response to a write failure on a second server. The Examiner furthermore believes that Blumenau's paragraph [0042] discloses that when the write is not completed, a migration process is restarted, pointing to Blumenau's statement that the "DBMS will reissue their request".

However, Blumenau's paragraph [0042] cannot be read out of context. In particular, in the immediate prior paragraph [0041] it is discussed that in view of the fact that the source volumes are maintained on line, there is a risk that the system might crash while one or more write operations are pending. It is then described that there are four possible states in which the write operation could possibly be when the system crashed.

In the first part of paragraph [0042] it is explained that in a first state (state 1), the write operation has not been performed successfully on either the source volume or the target volume. This state is believed by Blumenau to "not be problematic" as the source (B1) and target (B2) volumes are consistent, such that the migration is "not at risk of being performed inaccurately". Thus Blumenau says that in this state (and only in this state) it is entirely "within the control of the application program that issued the outstanding write request" to recover from the crash. This is because the application will not have received any acknowledgment that the write completed, and therefore the application can simply reissue the write request. In the case he

describes, where a database management server (DBMS) has issued the write request, the DBMS itself will keep a queue of operations and their status. Thus, there is no need for the lower level storage server to do so. If an operation is not completed within a certain amount of time, the DBMS will re-issue the request.

This section of Blumenau is thus referring to the actions of a high level application (e.g., a DBMS) when an operation is not completed within a certain amount of time. More importantly, this function of Blumenau's system is happening in a context where the two volumes are in no danger of being inconsistent. This situation is thus different from the claimed element of a write-detect process that responds to a write failure on a second server – a specific situation where there is very much the danger of the two volumes being inconsistent.

This teaching of Blumenau also does not amount to teaching that the very same migration process itself is detecting a write failure on a target volume and in response to that write failure, the migration process is restarting itself. Indeed, all that Blumenau suggests is that a higher, application-level DBMS operation will be reissued when no acknowledgement of its completion is received. The failure to acknowledge a DBMS operation can be due to a myriad of reasons and does not suggest in any way that a volume migration process will be restarted.

Indeed, Blumenau actually teaches away from Applicants' claimed invention. In later paragraphs [0046 through 0047] it is stated that because the migration is maintained to be transparent to the application, the application (e.g., his DBMS application) will not check the target volume to avoid inconsistency. Blumenau then goes on to explain how information in other states will be used to restore good data to the volume that is determined to be the bad. In other words, Blumenau teaches one to pick up from where the error occurred--and does not restart the migration process.

With respect to other aspects of the Applicants' Claim 1, we note that Blumenau also does not actually disclose that server processes communicate with each other using information in a state table to generate a measure of loading on respective servers. All that Blumenau's state table 105 stores is information concerning back up, copy and recovery. These features do not include, suggest, or teach maintaining data indicative of load across a plurality of servers, or doing anything in response to the same.

There are additional reasons why at least Applicants' Claim 6 should be allowed. The Examiner is of the opinion that Blumenau further discloses that a load monitor process determines whether a server is servicing a disproportionate share of client requests in a server group, referring to Blumenau's paragraph [0064]. There is a mention in that paragraph of monitoring when the storage system 402a becomes processor bound, approaching its performance limit, or storage capacity (e.g., storage system 402a may have one or more storage volumes that approach full capacity). But this does not amount to a teaching, suggestion or even an inference that there is any determination of the number of client requests or share of client requests being handled by a specific server.

We also believe that at least Claim 8 should be allowed. Blumenau admittedly does disclose a state information table that is used to track the state of a storage element in a logical volume 303. For example, in paragraph [0051] is explained that state information 301 may include a count 302 which indicates a number of data operations performed on a corresponding storage element of volume 303. It is also stated in paragraph [0054] that state information 301 may include other information used for recovery, such as to track information as needed for disk mirror processes. However, there is no mention, suggestion, or teaching in Blumenau that this state table is used to perform any routing function, or maintain data that would permit routing based on state.

Applicants' Claim 8 on the other hand, is directed to a routing table for tracking which resources are maintained on which servers in the system. Indeed, since Blumenau does not even disclose a plurality of storage servers that share a set of resources partitioned thereon, there is no need for him to either maintain a routing table or determine how to route client requests among such servers.

The Examiner also appears to make reference to paragraph [0064] in Blumenau as teaching that a load monitor process monitors one or more of network traffic load, I/O request load or storage traffic pattern type. But the Examiner is merely repeating Applicants' claim language. None of these functions are actually disclosed in Blumenau, which merely determines when a storage system becomes processor bound, approaches a performance limit, or is reaching storage capacity limits. These do not amount to teaching the more specifically claimed monitoring of traffic load, I/O request load, or storage traffic pattern type.

CONCLUSION

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By _____

David J. Thibodeau, Jr.
Registration No. 31,671
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133
Date: 6/4/08